Data Acquisition

PCI DAQB

DAQBench™ 32-bit ActiveX Controls for Measurement and Automation for Windows NT

Software Model Name: "DAQBench"

Motion Controls Reference

REVISION 2.20 — JUNE 15, 2001

No part of this manual may be reproduced without permission.

CyberResearch, Inc.

www.cyberresearch.com

25 Business Park Drive, Branford, CT 06405 USA

203-483-8815 (9am to 5pm EST) FAX: 203-483-9024

PCI DAQB

DAQBench™ 32-bit ActiveX Controls for Measurement and Automation for Windows NT

Software Model Name: “DAQBench”

Motion Controls Reference
Revision 2.20 – JUNE 15, 2001

No part of this manual may be reproduced without permission.

CyberResearch, Inc.
www.cyberresearch.com
25 Business Park Drive, Branford, CT 06405 USA
203-483-8815 (9am to 5pm EST) FAX: 203-483-9024

©Copyright 2001
All Rights Reserved.

Revision 2.20: May 15, 2001

The information in this document is subject to change without prior notice in order to improve reliability, design, and function and does not represent a commitment on the part of CyberResearch, Inc.

In no event will CyberResearch, Inc. be liable for direct, indirect, special, incidental, or consequential damages arising out of the use of or inability to use the product or documentation, even if advised of the possibility of such damages.

This document contains proprietary information protected by copyright. All rights are reserved. No part of this manual may be reproduced by any mechanical, electronic, or other means in any form without prior written permission of CyberResearch, Inc.

Trademarks

"CyberResearch" and "PCI DAQB" are trademarks of CyberResearch, Inc. Other product names mentioned herein are used for identification purposes only and may be trademarks and/or registered trademarks of their respective companies.

• NOTICE •

CyberResearch, Inc. does not authorize any CyberResearch product for use in life support systems, medical equipment, and/or medical devices without the written approval of the President of CyberResearch, Inc. Life support devices and systems are devices or systems which are intended for surgical implantation into the body, or to support or sustain life and whose failure to perform can be reasonably expected to result in injury. Other medical equipment includes devices used for monitoring, data acquisition, modification, or notification purposes in relation to life support, life sustaining, or vital statistic recording. CyberResearch products are not designed with the components required, are not subject to the testing required, and are not submitted to the certification required to ensure a level of reliability appropriate for the treatment and diagnosis of humans.

Contents

PCIM 1002E ActiveX Control 1
Properties 1
Methods 6
Events 13
PCIM 1004E ActiveX Control 14
Properties 14
Methods 19
Events 25
PCIM 2004E ActiveX Control 26
Properties 26
Methods 33
Events 47

1

PCIM 1002E ActiveX Control

The PCIM 1002E is a 2-axis, PCI-bus motion control card. The PCIM 1002E ActiveX control is a software component that provides the interface for users to control PCIM 1002E card.

Properties

Axis0 Property

Returns a object that is a PCIM 1002E Axis#.

Syntax

object.Axis0

Remarks

This property is read only.

Data Type

PCIM 1002E AXIS

Axis1 Property

Returns a object that is a PCIM 1002E Axis1.

Syntax

object.Axis1

Remarks

This property is read only.

Data Type

PCIM 1002E AXIS

Axis#.AlarmMode Property

Returns/sets alarm logic and alarm mode for one axis. The '#' can be '0' or '1'. It represents the axis identification.

Syntax

object.Axis#.AlarmMode [=number]

Settings

Value	Description
0	active low stop immediately
1	active high stop immediately
2	active low decelerate then stop
3	active high decelerate then stop

Data Type

Integer

Axis#.ErcStatus Property

Returns/sets ERC pin output enable/disable for one axis. The '#' can be '0' or '1'. It represents the axis identification.

Syntax

object.Axis#.ErcStatus [=number]

Settings

Value	Description
0	Disable
1	Enable

Data Type

Integer.

Axis#.EzStatus Property

Returns/sets index signal needed for home_move for one axis. The '#' can be '0' or '1'. It represents the axis identification.

Syntax

object.Axis#.EzStatus [=number]

Settings

Value	Description
0	active low
1	active high

Data Type

Integer

Axis#.HomeMode Property

Returns/sets the mode for home return for one axis. The '#' can be '0' or '1'. It represents the axis identification.

Syntax

object.Axis#.HomeMode [=number]

Settings

Value	Description
0	ORG only
1	ORG and EZ(stop)
2	ORG and EZ(decelerate then stop)

Data Type

Integer

Axis#.InpMode Property

Returns/sets In-Position logic and enable/disable for one axis. The '#' can be '0' or '1'. It represents the axis identification.

Syntax

object.Axis#.InpMode [=number]

Settings

Value	Description
0	disable
1	enable active low
2	enable active high

Data Type

Integer

Axis#.Inputmode Property

Returns/sets 'Set the mode for feedback pulse input' for one axis. The '#' can be '0' or '1'. It represents the axis identification.

Syntax

object.Axis#.Inputmode[=Integer]

Settings

Value	Description
0	1X AB PHASE
1	2X AB PHASE
2	4X AB PHASE
3	CW/CCW

Data Type

Integer

Axis#.InputStatus Property

Returns/sets enable/disable for the external feedback pulse input for one axis. The '#' can be '0' or '1'. It represents the axis identification.

Syntax

object.Axis#.InputStatus [=number]

Settings

Value	Description
0	Disable
1	Enable

Data Type

Integer.

Axis#.IntFactor Property

Returns/sets the 32-bit interrupt factor for one axis. The '#' can be '0' or '1'. It represents the axis identification.

Syntax

object.Axis#.IntFactor [=variant]

Settings

Bit	Interrupt Factor
0	Stop with the EL signal
1	Stop with the SD signal
2	Stop with the ALM signal
3	Stop with the STP signal
4	Should be set to 0
5	Completion of home return
6	Completion of preset movement
7	Completion of interpolating motion for two axes: (X & Y) or (Z & U)
8~12	_(should be set to 0)
13	when ***v_stop()*** function stop the axis
14	EA/EB, PA/PB encoder input error
15	start with STA signal
16	Completion of acceleration
17	Start of deceleration
18~22	Should be Set to 0
23	RDY active (AP3 of PCL5023 change from 1 to 0)
24~31	Should be set to 0

Note: Bit 14: The interrupt is generated when pins EA and EB, or PA and PB change simultaneously. It means there is an encoder input error.

Remarks

In VC++,C++ Builder
Variant is a VARIANT of VT_I4.

Data Type

VARIANT.

Axis#.IOStatus Property

Returns all the I/O status for one axis. The '#' can be '0' or '1'. It represents the axis identification.

Syntax

object.Axis#.IOStatus [=variant]

Settings

Bit	Name	Description
0	+EL	Positive Limit Switch
1	-EL	Negative Limit Switch
2	+SD	Positive Slow Down Point
3	-SD	Negative Slow Down Point
4	ORG	Origin Switch
5	EZ	Index signal
6	ALM	Alarm Signal
7	SVON	SVON of PCL5023 pin output
8	RDY	RDY pin input
9	INT	Interrupt status
10	ERC	ERC pin output
11	INP	In-Position signal input

Remarks

This property is read only.
In VC++,C++ Builder
Variant is a VARIANT of VT_I2.

Data Type

VARIANT

Axis#.ManualInputMode Property

Returns/sets pulser input mode and operation mode for one axis. The '#' can be '0' or '1'. It represents the axis identification.

Syntax

object.Axis#.ManualInputMode [=number]

Settings

Value	Description
0	1X AB PHASE Independent Mode
1	2X AB PHASE Independent Mode
2	4X AB PHASE Independent Mode
3	CW/CCW Independent Mode
4	1X AB PHASE Common Mode
5	2X AB PHASE Common Mode
6	4X AB PHASE Common Mode
7	CW/CCW Common Mode

Data Type

Integer.

Axis#.OrgMode Property

Returns/sets Configure the logic of origin switch for one axis. The '#' can be '0' or '1'. It represents the axis identification.

Syntax

object.Axis#.OrgMode [=number]

Settings

Value	Description
0	active low NOT latch
1	active high NOT latch
2	active low AND latch
3	active high AND latch

Data Type

Integer

Axis#.Outputmode Property

Returns/sets the mode of pulse command output for one axis. The '#' can be '0' or '1'. It represents the axis identification.

Syntax

object. Axis#.Outputmode[=Integer]

Settings

Value	Description
0	OUT_DIR
1	CW_CCW

Data Type

Integer

Axis#.ScalingFactor Property

Returns/sets resolution ratio for one axis. The '#' can be '0' or '1'. It represents the axis identification.

Syntax

object.Axis#.ScalingFactor [=value]

Settings

resolution ratio for the axis.

Data Type

Double

Axis#.SdMode Property

Returns/sets active logic and latch control of SD signal input from mechanical system for one axis. The '#' can be '0' or '1'. It represents the axis identification.

Syntax

object.Axis#.SdMode [=number]

Settings

Value	Description
0	disable
1	enable active low NOT latch
2	enable active high NOT latch
3	enable active low latch
4	enable active high latch

Data Type

Integer

Axis#.Svon Property

Returns/sets the High/Low state of general purpose output pin SVON for one axis. The '#' can be '0' or '1'. It represents the axis identification.

Syntax

object.Axis#.Svon [=number]

Settings

Value	Description
0	SVON LOW
1	SVON HIGH

Data Type

Integer.

DeviceIndex Property

Returns/sets a value that determines the card index of PCIM 1002E .

Syntax

object. DeviceIndex [= number]

Data Type

Integer.

IntStatus Property

Returns/sets the status of interrupt.

Syntax

object. IntStatus [= Integer]

Settings

Value	Description
1	enable interrupt.
0	disable interrupt.

Data Type

Integer

LoadConfig Property

Returns/sets a value that determines the mode of opening device .

Syntax

object. LoadConfig [= Boolean]

Settings

Value	Description
True	Load the same configuration as “Motion Creator”.
False	Don’t load the configuration form “Motion Creator”.

Data Type

Boolean

OpenMode Property

Returns/sets a value that determines the mode of opening device .

Syntax

object.**OpenMode** [= number]

Settings

Value	Description
0	Automatically open device when the control was created
1	Don’t open device when the control was created. Must call the Open method to open device. (Manual)

Data Type

Integer.

Methods

BuildCompareTable Method

Syntax

Function *object*.BuildCompareTable (axis As Integer, table as variant, size as Integer) As Integer

Arguments

axis As Integer : axis number designated to move or stop.

size As Integer : the element’s number you want to build in the compare table.

Table as Variant : compare table

In VC++, C++ Builder

Table is a variant of VT_ARRAY|VT_I4.

Return Value

0 if the function is successful.

Remarks

Build compare table

GetDI Method

Syntax

Function *object*.GetDI (DiData As variant) As Integer

Arguments

DiData As Variant : a 16-bit input value

In VC++, C++ Builder

DiData is a variant(VT_I2) address to save data.

Return Value

0 if the function is successful.

Remarks

Gets a 16-bit value from PCIM 1002E's digital input channel.

GetPosition Method

Syntax

Function *object***.** GetPosition (axis As Integer, pos As Double) As Integer

Arguments

axis As Integer : axis number designated to move or stop.

pos As Double : actual position

Return Value

0 if the function is successful.

Remarks

get the actual position
In VC++,C++ Builder
Pos is the address to save the position.

HomeMove Method

Syntax

Function *object***.**HomeMove (axis As Integer, svel As Double, mvel As Double, accel As Double) As Integer

Arguments

axis As Integer : axis number designated to move or stop.

svel As Double : starting velocity in units of pulse per second

mvel As Double : maximum velocity in units of pulse per second

accel As Double : specified acceleration time in units of seconds.

Return Value

0 if the function is successful.

Remarks

Begin a home return move

Open Method

Syntax

Function *object***.Open (***[ErrMsgBox As Variant]***)** As Boolean

Arguments

[ErrMsgBox As Variant]
It is optional and boolean type. Default value is False
True, It will popup error message dialog box when the opening device is failed.
False, It will fire DAQError event instead of popping up dialog when the opening device is failed.

Return Value

True if the function is successful; otherwise False.

Remarks

This method will be use when the OpenMode property is Manual.

Note

In VC++, *ErrMsgBox* is a VARIANT of VT_I2.

SetCompareHome Method

Syntax

Function *object*.SetCompareHome (axis As Integer) As Integer

Arguments

axis As Integer : axis number designated to move or stop.

Return Value

0 if the function is successful.

Remarks

Reset the comparator's counter to zero

SetCompareInt Method

Syntax

Function *object*.SetCompareInt (axis As Integer, status as Integer) As Integer

Arguments

axis As Integer : axis number designated to move or stop.

status As Integer : enable/disable the compare interrupt.

1 means enable,

0 means disable

Return Value

0 if the function is successful.

Remarks

Enable/disable the compare interrupt

SetCompareMode Method

Syntax

Function *object*.SetCompareMode (axis As Integer, comp_mode as Integer) As Integer

Arguments

axis As Integer : axis number designated to move or stop.

Comp_mode As Integer : compare mode.

0=increasing (counter > compare value)
1=equal (counter = compare value)
2=decreasing (counter < compare value)

Return Value

0 if the function is successful.

Remarks

set compare mode

SetCompareTable Method

Syntax

Function *object*.SetCompareTable (axis As Integer, control as Integer) As Integer

Arguments

axis As Integer : axis number designated to move or stop.

control As Integer : enable/disable compare table.

0 means disable
1 means compare points is from compare table

Return Value

0 if the function is successful.

Remarks

Enable/disable the compare table

SetConfigFlie Method

Syntax

Function *object*.SetConfigFlie (filename As String) As Integer

Arguments

filename As String : the configuration file name.

Return Value

0 if the function is successful.

Remarks

Save the configuration

SetDO Method

Syntax

Function *object*.SetDO (DoData As variant) As Integer

Arguments

DoData As Variant : a 16-bit output value

In VC++, C++ Builder

Table is a variant of VT_I2.

Return Value

0 if the function is successful.

Remarks

Set a 16-bit value to PCIM 1002E's digital output channel.

SetPosition Method

Syntax

Function *object*. SetPosition (axis As Integer, pos As Double) As Integer

Arguments

axis As Integer : axis number designated to move or stop.

pos As Double : actual position

Return Value

0 if the function is successful.

Remarks

Set the actual position

StartAMove Method

Syntax

Function *object*.StartAMove (axis As Integer, pos As Double, str_vel As Double, max_vel As Double, Tacc As Double) As Integer

Arguments

axis As Integer : axis number designated to move or stop.

Pos As Double : the absolute position to move.

str_vel As Double : starting velocity in units of pulse per second.

max_vel As Double : maximum velocity in units of pulse per second.

Tacc As Double : specified acceleration time in units of seconds.

Return Value

0 if the function is successful.

Remarks

This function causes the axis to accelerate from a starting velocity.

StartManualMove Method

Syntax

Function *object*. StartManualMove (axis As Integer) As Integer

Arguments

axis As Integer : axis number designated to move or stop.

Return Value

0 if the function is successful.

Remarks

Begin a manual pulser movement.

StartMoveXY Method

Syntax

Function *object*.StartMoveXY (x As Double, y As Double, str_vel As Double, max_vel As Double, accel As Double) As Integer

Arguments

x, y As Double : the absolute position to move.

str_vel As Double : starting velocity in units of pulse per second.

max_vel As Double : maximum velocity in units of pulse per second.

accel As Double : specified acceleration time in units of seconds.

Return Value

0 if the function is successful.

Remarks

Perform a 2-axis linear interpolated motion between X & Y.

StartRMove Method

Syntax

Function *object*.StartRMove (axis As Integer, distance As Double, str_vel As Double, max_vel As Double, Tacc As Double) As Integer

Arguments

axis As Integer : axis number designated to move or stop.

distance As Double : the relative distance to move.

str_vel As Double : starting velocity in units of pulse per second.

max_vel As Double : maximum velocity in units of pulse per second.

Tacc As Double : specified acceleration time in units of seconds.

Return Value

0 if the function is successful.

Remarks

Begin a relative trapezoidal profile move.

StartRSMove Method

Syntax

Function *object*.StartRSMove (axis As Integer, distance As Double, str_vel As Double, max_vel As Double, Tlacc As Double, Tsacc As Double) As Integer

Arguments

axis As Integer : axis number designated to move or stop.

disatnce As Double : the relative distance to move.

str_vel As Double : starting velocity in units of pulse per second.

max_vel As Double : maximum velocity in units of pulse per second.

Tlacc, Tsacc As Double : specified acceleration time in units of seconds.

Return Value

0 if the function is successful.

Remarks

Begin a relative S-Curve profile motion.

StartSMove Method

Syntax

Function *object*.StartSMove (axis As Integer, pos As Double, str_vel As Double, max_vel As Double, Tlacc As Double, Tsacc As Double) As Integer

Arguments

axis As Integer : axis number designated to move or stop.

pos As Double : the absolute position to move.

str_vel As Double : starting velocity in units of pulse per second.

max_vel As Double : maximum velocity in units of pulse per second.

Tlacc, Tsacc As Double : specified acceleration time in units of seconds.

Return Value

0 if the function is successful.

Remarks

Begin a S-Curve profile motion.

StartTAMove Method

Syntax

Function *object*.StartTAMove (axis As Integer, pos As Double, str_vel As Double, max_vel As Double, Tacc As Double, Tdec As Double) As Integer

Arguments

axis As Integer : axis number designated to move or stop.

pos As Double : the absolute position to move.

str_vel As Double : starting velocity in units of pulse per second.

max_vel As Double : maximum velocity in units of pulse per second.

Tacc, Tdec As Double : specified acceleration time in units of seconds.

Return Value

0 if the function is successful.

Remarks

Begin a non-symmetrical absolute trapezoidal profile move.

StartTASMove Method

Syntax

Function *object*.StartTASMove (axis As Integer, pos As Double, str_vel As Double, max_vel As Double, Tlacc As Double, Tsacc As Double, Tlacc As Double, Tsdec As Double) As Integer

Arguments

axis As Integer : axis number designated to move or stop.

pos As Double : the absolute position to move.

str_vel As Double : starting velocity in units of pulse per second.

max_vel As Double : maximum velocity in units of pulse per second.

Tlacc, Tsacc, Tlacc, Tsdec As Double : specified acceleration time in units of seconds.

Return Value

0 if the function is successful.

Remarks

Begin a non-symmetrical absolute S-curve profile motion.

StartTMove Method

Syntax

Function *object*.StartTMove (axis As Integer, distance As Double, str_vel As Double, max_vel As Double, Tacc As Double, Tdec As Double) As Integer

Arguments

axis As Integer : axis number designated to move or stop.

distance As Double : the relative distance to move.

str_vel As Double : starting velocity in units of pulse per second.

max_vel As Double : maximum velocity in units of pulse per second.

Tacc, Tdec As Double : specified acceleration time in units of seconds.

Return Value

0 if the function is successful.

Remarks

Begin a non-symmetrical relative trapezoidal profile move.

SVMove Method

Syntax

Function *object*.SVMove(axis As Integer, str_vel As Double, max_vel As Double, Tlacc As Double, Tsacc As Double) As short

Arguments

axis As Integer : axis number designated to move or stop.

0	axis 0
1	axis 1

str_vel As Double : starting velocity in units of pulse per second.

max_vel As Double : maximum velocity in units of pulse per second.

Tsacc, Tlacc As Double : specified acceleration time in units of seconds.

Return Value

0 if the function is successful.

Remarks

This method will accelerate an axis to a constant velocity with S-curve profile.

VChange Method

Syntax

Function *object*.VChange(axis As Integer, max_vel As Double, Tlacc As Double) As Integer

Arguments

axis As Integer : axis number designated to move or stop.

max_vel As Double : maximum velocity in units of pulse per second.

Tlacc As Double : specified acceleration time in units of seconds.

Return Value

0 if the function is successful.

Remarks

This method will Change speed instantly.

VMove Method

Syntax

Function *object*.VMove(axis As Integer, str_vel As Double, max_vel As Double, Tlacc As Double) As Integer

Arguments

axis As Integer : axis number designated to move or stop.

str_vel As Double : starting velocity in units of pulse per second.

max_vel As Double : maximum velocity in units of pulse per second.

Tlacc As Double : specified acceleration time in units of seconds.

Return Value

0 if the function is successful.

Remarks

This method will accelerate an axis to a constant velocity with T-curve profile.

VStop Method

Syntax

Function *object*.VStop(axis As Integer Tdec As Double) As Integer

Arguments

axis As Integer : axis number designated to move or stop.

Tdec As Double : specified acceleration time in units of seconds.

Return Value

0 if the function is successful.

Remarks

This method will make axis decelerate to a stop.

Events

OnIntEvent Event

Syntax

sub *ControlName_* OnIntEvent**(** *axis As Integer, Int_status As Long* **)**

Arguments

axis As Intege : the axis that generates an interrupt.
Int_status As Long : the interrupt status.

Remarks

This event will occur when an interrupt is generated.

Note

User must first enable the interrupt and set the interrupt factor, then this event can occur.

2

PCIM 1004E ActiveX Control

The PCIM 1004E is a 4-axis, PCI-bus motion control card. The PCIM 1004E ActiveX control is a software component that provides the interface for users to control PCIM 1004E card.

Properties

Axis0 Property

Returns a object that is a PCIM 1004E Axis#.

Syntax

object.Axis0

Remarks

This property is read only.

Data Type

PCIM 1004E AXIS

Axis1 Property

Returns a object that is a PCIM 1004E Axis1.

Syntax

object.Axis1

Remarks

This property is read only.

Data Type

PCIM 1004E AXIS

Axis2 Property

Returns a object that is a PCIM 1004E Axis2.

Syntax

object.Axis0

Remarks

This property is read only.

Data Type

PCIM 1004E AXIS

Axis3 Property

Returns a object that is a PCIM 1004E Axis3.

Syntax

object.Axis1

Remarks

This property is read only.

Data Type

PCIM 1004E AXIS

Axis#.AlarmMode Property

Returns/sets alarm logic and alarm mode for one axis. The '#' can be '0' to '3'. It represents the axis identification.

Syntax

object.Axis#.AlarmMode [=number]

Settings

Value	Description
0	active low stop immediately
1	active high stop immediately
2	active low decelerate then stop
3	active high decelerate then stop

Data Type

Integer.

Axis#.ErcStatus Property

Returns/sets ERC pin output enable/disable for one axis. The '#' can be '0' to '3'. It represents the axis identification.

Syntax

object.Axis#.ErcStatus [=number]

Settings

Value	Description
0	Disable
1	Enable

Data Type

Integer.

Axis#.EzStatus Property

Returns/sets index signal needed for home_move for one axis. The '#' can be '0' to '3'. It represents the axis identification.

Syntax

object.Axis#.EzStatus [=number]

Settings

Value	Description
0	active low
1	active high

Data Type

Integer.

Axis#.HomeMode Property

Returns/sets the mode for home return for one axis. The '#' can be '0' to '3'. It represents the axis identification.

Syntax

object.Axis#.HomeMode [=number]

Settings

Value	Description
0	ORG only
1	ORG and EZ(stop)
2	ORG and EZ(decelerate then stop)

Data Type

Integer.

Axis#.InpMode Property

Returns/sets In-Position logic and enable/disable for one axis. The '#' can be '0' to '3'. It represents the axis identification.

Syntax

object.Axis#.InpMode [=number]

Settings

Value	Description
0	disable
1	enable active low
2	enable active high

Data Type

Integer.

Axis#.Inputmode Property

Returns/sets Set the mode for feedback pulse input for one axis. The '#' can be '0' to '3'. It represents the axis identification.

Syntax

object.Axis#.Inputmode[=Integer]

Settings

Value	Description
0	1X AB PHASE
1	2X AB PHASE
2	4X AB PHASE
3	CW/CCW

Data Type

Integer.

Axis#.InputStatus Property

Returns/sets enable/disable the external feedback pulse input for one axis. The '#' can be '0' to '3'. It represents the axis identification.

Syntax

object.Axis#.InputStatus [=number]

Settings

Value	Description
0	Disable
1	Enable

Data Type

Integer.

Axis#.IntFactor Property

Returns/sets the 32-bit interrupt factor for one axis. The '#' can be '0' to '3'. It represents the axis identification.

Syntax

object.Axis#.IntFactor [=variant]

Settings

Bit	Interrupt Factor
0	Stop with the EL signal
1	Stop with the SD signal
2	Stop with the ALM signal
3	Stop with the STP signal
4	Should be set to 0
5	Completion of home return
6	Completion of preset movement
7	Completion of interpolating motion for two axes: (X & Y) or (Z & U)
8~12	_(should be set to 0)
13	when ***v_stop()*** function stop the axis
14	EA/EB, PA/PB encoder input error
15	start with STA signal
16	Completion of acceleration
17	Start of deceleration

18~22	Should be Set to 0
23	RDY active (AP3 of PCL5023 change from 1 to 0)
24~31	Should be set to 0

Note: Bit 14: The interrupt is generated when pins EA and EB, or PA and PB change simultaneously. It means there is an encoder input error.

Remarks

In VC++,C++ Builder
variant is a VARIANT of VT_I4.

Data Type

VARIANT.

Axis#.IOStatus Property

Returns all the I/O status for one axis. The '#' can be '0' to '3'. It represents the axis identification.

Syntax

object.Axis#.IOStatus [=variant]

Settings

Bit	Name	Description
0	+EL	Positive Limit Switch
1	-EL	Negative Limit Switch
2	+SD	Positive Slow Down Point
3	-SD	Negative Slow Down Point
4	ORG	Origin Switch
5	EZ	Index signal
6	ALM	Alarm Signal
7	SVON	SVON of PCL5023 pin output
8	RDY	RDY pin input
9	INT	Interrupt status
10	ERC	ERC pin output
11	INP	In-Position signal input

Remarks

This property is read only.
In VC++,C++ Builder
variant is a VARIANT of VT_I2.

Data Type

VARIANT.

Axis#.ManualInputMode Property

Returns/sets pulser input mode and operation mode for one axis. The '#' can be '0' to '3'. It represents the axis identification.

Syntax

object.Axis#.ManualInputMode [=number]

Settings

Value	Description
8	1X AB PHASE Independent Mode
9	2X AB PHASE Independent Mode
10	4X AB PHASE Independent Mode
11	CW/CCW Independent Mode
12	1X AB PHASE Common Mode
13	2X AB PHASE Common Mode
14	4X AB PHASE Common Mode
15	CW/CCW Common Mode

Data Type

Integer.

Axis#.OrgMode Property

Returns/sets Configure the logic of origin switch for one axis. The '#' can be '0' to '3'. It represents the axis identification.

Syntax

object.Axis#.OrgMode [=number]

Settings

Value	Description
0	active low not latch
1	active high not latch
2	active low and latch
3	active high and latch

Data Type

Integer.

Axis#.Outputmode Property

Returns/sets Set the mode for pulse command output for one axis. The '#' can be '0' to '3'. It represents the axis identification.

Syntax

object. Axis#.Outputmode[=Integer]

Settings

Value	Description
0	OUT_DIR
1	CW_CCW

Data Type

Integer.

Axis#.ScalingFactor Property

Returns/sets resolution ratio for one axis. The '#' can be '0' to '3'. It represents the axis identification.

Syntax

object.Axis#.ScalingFactor [=value]

Settings

Resolution ratio for the axis.

Data Type

Double.

Axis#.SdMode Property

Returns/sets active logic and latch control of SD signal input from mechanical system for one axis. The '#' can be '0' to '3'. It represents the axis identification.

Syntax

object.Axis#.SdMode [=number]

Settings

Value	Description
0	disable
1	enable active low not latch
2	enable active high not latch
3	enable active low latch
4	enable active high latch

Data Type

Integer.

Axis#.Svon Property

Returns/sets the High/Low state of general purpose output pin SVON for one axis. The '#' can be '0' to '3'. It represents the axis identification.

Syntax

object.Axis#.Svon [=number]

Settings

Value	Description
0	SVON LOW
1	SVON HIGH

Data Type

Integer.

DeviceIndex Property

Returns/sets a value that determines the card index of PCIM 1002E .

Syntax

object. DeviceIndex [= number]

Data Type

Integer.

IntStatus Property

Returns/sets the status of interrupt.

Syntax

object. IntStatus [= Integer]

Settings

Value	Description
1	enable interrupt.
0	disable interrupt.

Data Type

Integer.

LoadConfig Property

Returns/sets a value that determines the mode of opening the device .

Syntax

object. LoadConfig [= Boolean]

Settings

Value	Description
True	Load the same configuration as “Motion Creator”.
False	Do not load the configuration from “Motion Creator”.

Data Type

Boolean.

OpenMode Property

Returns/sets a value that determines the mode of opening the device .

Syntax

object.**OpenMode** [= number]

Settings

Value	Description
0	Automatically open device when the control was created.
1	Do not open device when the control was created. Must call the Open method to open device. (Manual)

Data Type

Integer.

Methods

GetPosition Method

Syntax

Function *object***.** GetPosition (axis As Integer, pos As Double) As Integer

Arguments

axis As Integer : axis number designated to move or stop.

pos As Double : actual position

Return Value

0 if the function is successful.

Remarks

Get the actual position in VC++,C++ Builder.
Pos is the address to save the position.

HomeMove Method

Syntax

Function *object***.**HomeMove (axis As Integer, svel As Double, mvel As Double, accel As Double) As Integer

Arguments

axis As Integer : axis number designated to move or stop.

svel As Double : starting velocity in units of pulse per second

mvel As Double : maximum velocity in units of pulse per second

accel As Double : specified acceleration time in units of seconds.

Return Value

0 if the function is successful.

Remarks

Begin a home return move.

Open Method

Syntax

Function *object***.Open (***[ErrMsgBox As Variant]***)** As Boolean

Arguments

[ErrMsgBox As Variant]
It is optional and boolean type. Default value is False.
True, It will popup an error message dialog box when the opening device is failed.
False, It will give a DAQError event instead of popping up dialog when the opening device is failed.

Return Value

True if the function is successful; otherwise False.

Remarks

This method will be used when the OpenMode property is Manual.

Note

In VC++, *ErrMsgBox* is a VARIANT of VT_I2.

SetConfigFlie Method

Syntax

Function *object***.**SetConfigFlie (filename As string) As Integer

Arguments

filename As String : the configuration file name.

Return Value

0 if the function is successful.

Remarks

Save the configuration.

SetPosition Method

Syntax

Function *object*. SetPosition (axis As Integer, pos As Double) As Integer

Arguments

axis As Integer : axis number designated to move or stop.

pos As Double : actual position.

Return Value

0 if the function is successful.

Remarks

Set the actual position.

StartAMove Method

Syntax

Function *object*.StartAMove (axis As Integer, pos As Double, str_vel As Double, max_vel As Double, Tacc As Double) As Integer

Arguments

axis As Integer : axis number designated to move or stop.

Pos As Double : the absolute position to move.

str_vel As Double : starting velocity in units of pulse per second.

max_vel As Double : maximum velocity in units of pulse per second.

Tacc As Double : specified acceleration time in units of seconds.

Return Value

0 if the function is successful.

Remarks

This function causes the axis to accelerate from a starting velocity.

StartManualMove Method

Syntax

Function *object*. StartManualMove (axis As Integer) As Integer

Arguments

axis As Integer : axis number designated to move or stop.

Return Value

0 if the function is successful.

Remarks

Begin a manual pulser movement.

StartMoveXY Method

Syntax

Function *object*.StartMoveXY (x As Double, y As Double, str_vel As Double, max_vel As Double, accel As Double) As Integer

Arguments

x, y As Double : the absolute position to move.

str_vel As Double : starting velocity in units of pulse per second.

max_vel As Double : maximum velocity in units of pulse per second.

accel As Double : specified acceleration time in units of seconds.

Return Value

0 if the function is successful.

Remarks

Perform a 2-axis linear interpolated motion between X & Y.

StartMoveZU Method

Syntax

Function *object*.StartMoveXY (z As Double, u As Double, str_vel As Double, max_vel As Double, accel As Double) As Integer

Arguments

z, u As Double : the absolute position to move.

str_vel As Double : starting velocity in units of pulse per second.

max_vel As Double : maximum velocity in units of pulse per second.

accel As Double : specified acceleration time in units of seconds.

Return Value

0 if the function is successful.

Remarks

Perform a 2-axis linear interpolated motion between Z & U.

StartRMove Method

Syntax

Function *object*.StartRMove (axis As Integer, distance As Double, str_vel As Double, max_vel As Double, Tacc As Double) As Integer

Arguments

axis As Integer : axis number designated to move or stop.

distance As Double : the relative distance to move.

str_vel As Double : starting velocity in units of pulse per second.

max_vel As Double : maximum velocity in units of pulse per second.

Tacc As Double : specified acceleration time in units of seconds.

Return Value

0 if the function is successful.

Remarks

Begin a relative trapezoidal profile move.

StartRSMove Method

Syntax

Function *object*.StartRSMove (axis As Integer, distance As Double, str_vel As Double, max_vel As Double, Tlacc As Double, Tsacc As Double) As Integer

Arguments

axis As Integer : axis number designated to move or stop.

disatnce As Double : the relative distance to move.

str_vel As Double : starting velocity in units of pulse per second.

max_vel As Double : maximum velocity in units of pulse per second.

Tlacc, Tsacc As Double : specified acceleration time in units of seconds.

Return Value

0 if the function is successful.

Remarks

Begin a relative S-Curve profile motion.

StartSMove Method

Syntax

Function *object*.StartSMove (axis As Integer, pos As Double, str_vel As Double, max_vel As Double, Tlacc As Double, Tsacc As Double) As Integer

Arguments

axis As Integer : axis number designated to move or stop.

pos As Double : the absolute position to move.

str_vel As Double : starting velocity in units of pulse per second.

max_vel As Double : maximum velocity in units of pulse per second.

Tlacc, Tsacc As Double : specified acceleration time in units of seconds.

Return Value

0 if the function is successful.

Remarks

Begin a S-Curve profile motion.

StartTAMove Method

Syntax

Function *object*.StartTAMove (axis As Integer, pos As Double, str_vel As Double, max_vel As Double, Tacc As Double, Tdec As Double) As Integer

Arguments

axis As Integer : axis number designated to move or stop.

pos As Double : the absolute position to move.

str_vel As Double : starting velocity in units of pulse per second.

max_vel As Double : maximum velocity in units of pulse per second.

Tacc, Tdec As Double : specified acceleration time in units of seconds.

Return Value

0 if the function is successful.

Remarks

Begin a non-symmetrical absolute trapezoidal profile move.

StartTASMove Method

Syntax

Function *object*.StartTASMove (axis As Integer, pos As Double, str_vel As Double, max_vel As Double, Tlacc As Double, Tsacc As Double, Tlacc As Double, Tsdec As Double) As Integer

Arguments

axis As Integer : axis number designated to move or stop.

pos As Double : the absolute position to move.

str_vel As Double : starting velocity in units of pulse per second.

max_vel As Double : maximum velocity in units of pulse per second.

Tlacc, Tsacc, Tlacc, Tsdec As Double : specified acceleration time in units of seconds.

Return Value

0 if the function is successful.

Remarks

Begin a non-symmetrical absolute S-curve profile motion.

StartTMove Method

Syntax

Function *object*.StartTMove (axis As Integer, distance As Double, str_vel As Double, max_vel As Double, Tacc As Double, Tdec As Double) As Integer

Arguments

axis As Integer : axis number designated to move or stop.

distance As Double : the relative distance to move.

str_vel As Double : starting velocity in units of pulse per second.

max_vel As Double : maximum velocity in units of pulse per second.

Tacc, Tdec As Double : specified acceleration time in units of seconds.

Return Value

0 if the function is successful.

Remarks

Begin a non-symmetrical relative trapezoidal profile move.

SVMove Method

Syntax

Function *object*.SVMove(axis As Integer, str_vel As Double, max_vel As Double, Tlacc As Double, Tsacc As Double) As short

Arguments

axis As Integer : axis number designated to move or stop.

str_vel As Double : starting velocity in units of pulse per second.

max_vel As Double : maximum velocity in units of pulse per second.

Tsacc, Tlacc As Double : specified acceleration time in units of seconds.

Return Value

0 if the function is successful.

Remarks

This method will accelerate an axis to a constant velocity with S-curve profile.

VChange Method

Syntax

Function *object*.VChange(axis As Integer, max_vel As Double, Tlacc As Double) As Integer

Arguments

axis As Integer : axis number designated to move or stop.

max_vel As Double : maximum velocity in units of pulse per second.

Tlacc As Double : specified acceleration time in units of seconds.

Return Value

0 if the function is successful.

Remarks

This method will Change speed instantly.

VMove Method

Syntax

Function *object*.VMove(axis As Integer, str_vel As Double, max_vel As Double, Tlacc As Double) As Integer

Arguments

axis As Integer : axis number designated to move or stop.

0	axis 0
1	axis 1
2	axis 2
3	axis3

str_vel As Double : starting velocity in units of pulse per second.

max_vel As Double : maximum velocity in units of pulse per second.

Tlacc As Double : specified acceleration time in units of seconds.

Return Value

0 if the function is successful.

Remarks

This method will accelerate an axis to a constant velocity with T-curve profile.

VStop Method

Syntax

Function *object*.VStop(axis As Integer Tdec As Double) As Integer

Arguments

axis As Integer : axis number designated to move or stop.

Tdec As Double : specified acceleration time in units of seconds.

Return Value

0 if the function is successful.

Remarks

This method will make the axis decelerate to a stop.

Events

OnIntEvent Event

Syntax

Sub *ControlName_* OnIntEvent**(** *axis As Integer, Int_status As Long* **)**

Arguments

axis As Integer : the axis that generates an interrupt.
Int_status As Long : the interrupt status.

Remarks

This event will occur when an interrupt is generated.

Note

User must first enable the interrupt and set the interrupt factor, then this event can occur.

3

PCIM 2004E ActiveX Control

The PCIM 2004E is a 4-axis, PCI-bus motion control card. The PCIM 2004E ActiveX control is a software component that provides the interface for users to control PCIM 2004E card.

Properties

Axis0 Property

Returns a object that is a PCIM 2004E Axis#.

Syntax

object.Axis0

Remarks

This property is read only.

Data Type

PCIM 2004E AXIS

Axis1 Property

Returns a object that is a PCIM 2004E Axis1.

Syntax

object.Axis1

Remarks

This property is read only.

Data Type

PCIM 2004E AXIS

Axis2 Property

Returns a object that is a PCIM 2004E Axis2.

Syntax

object.Axis0

Remarks

This property is read only.

Data Type

PCIM 2004E AXIS

Axis3 Property

Returns a object that is a PCIM 2004E Axis3.

Syntax

object.Axis1

Remarks

This property is read only.

Data Type

PCIM 2004E AXIS

Axis#.AlarmMode Property

Returns/sets alarm logic and alarm mode for one axis. The '#' can be '0' to '3'. It represents the axis identification.

Syntax

object.Axis#.AlarmMode [=number]

Settings

Value	Description
0	active low stop immediately
1	active high stop immediately
2	active low decelerate then stop
3	active high decelerate then stop

Data Type

Integer.

Axis#.ELMode Property

Returns/sets reacting modes when receiving EL signal for one axis. The '#' can be '0' to '3'. It represents the axis identification.

Syntax

object.Axis#.ELMode [=number]

Settings

Value	Description
0	stop
1	deceleration then stop

Data Type

Integer.

Axis#.ErcOut Property

Returns/sets ERC output option for one axis. The '#' can be '0' to '3'. It represents the axis identification.

Syntax

object.Axis#.ErcOut [=number]

Settings

Value	Description
0	no erc out
1	erc out when homing finish

Data Type

Integer.

Axis#.ErcStatus Property

Returns/sets ERC pin output enable/disable for one axis. The '#' can be '0' to '3'. It represents the axis identification.

Syntax

object.Axis#.ErcStatus [=number]

Settings

Value	Description
0	12us Active Low
1	102us Active Low
2	409us Active Low
3	1600us Active Low
4	13ms Active Low
5	52ms Active Low
6	104ms Active Low
7	12us Active High
8	102us Active High
9	409us Active High
10	1600us Active High
11	13ms Active High

12	52ms Active High
13	104ms Active High

Data Type

Integer.

Axis#.EzCount Property

Returns/sets EZ count for one axis. The '#' can be '0' to '3'. It represents the axis identification.

Syntax

object.Axis#.EzCount [=number]

Settings

Please refer to User's manual section 4.1.8

Data Type

Integer.

Axis#.EzStatus Property

Returns/sets index signal needed for home_move for one axis. The '#' can be '0' to '3'. It represents the axis identification.

Syntax

object.Axis#.EzStatus [=number]

Settings

Value	Description
0	active low
1	active high

Data Type

Integer.

Axis#.HomeMode Property

Returns/sets the mode for home return for one axis. The '#' can be '0' to '3'. It represents the axis identification.

Syntax

object.Axis#.HomeMode [=number]

Settings

Please refer to User's Manual section 4.1.8

Data Type

Integer.

Axis#.InpMode Property

Returns/sets In-Position logic and enable/disable for one axis. The '#' can be '0' to '3'. It represents the axis identification.

Syntax

object.Axis#.InpMode [=number]

Settings

Value	Description
0	disable
1	enable active low
2	enable active high

Data Type

Integer.

Axis#.Inputmode Property

Returns/sets Set the mode for feedback pulse input for one axis. The '#' can be '0' to '3'. It represents the axis identification.

Syntax

object.Axis#.Inputmode[=Integer]

Settings

Value	Description
0	1X AB PHASE Normal Low
1	2X AB PHASE Normal Low
2	4X AB PHASE Normal Low
3	CW/CCW Normal Low
4	1X AB PHASE Normal High
5	2X AB PHASE Normal High
6	4X AB PHASE Normal High
7	CW/CCW Normal High

Data Type

Integer.

Axis#.InputStatus Property

Returns/sets enable/disable the external feedback pulse input for one axis. The '#' can be '0' to '3'. It represents the axis identification.

Syntax

object.Axis#.InputStatus [=number]

Settings

Value	Description
0	External Feedback
1	Command pulse

Data Type

Integer.

Axis#.IntFactor Property

Returns/sets the 32-bit interrupt factor for one axis. The '#' can be '0' to '3'. It represents the axis identification.

Syntax

object.Axis#.IntFactor [=variant]

Settings

Bit	Interrupt Factor
0	Normal Stop
1	Next command continued
2	Continuous pre-register is empty
3	(Reserved)
4	Acceleration Start
5	Acceleration End
6	Deceleration Start
7	Deceleration End
8	(Reserved)
9	(Reserved)
10	Step-losing occur
11	General comparator compared
12	Compared triggered for axis 0,1
13	(Reserved)
14	Latched for axis2,3
15	ORG on
16	SD on
17~31	(Reserved)

Remarks

In VC++,C++ Builder
variant is a VARIANT of VT_I4.

Data Type

VARIANT.

Axis#.IOStatus Property

Returns all the I/O status for one axis. The '#' can be '0' to '3'. It represents the axis identification.

Syntax

object.Axis#.IOStatus [=variant]

Settings

Bit	Name	Description
0	RDY	RDY pin input
1	ALM	Alarm Signal
2	+EL	Positive Limit Switch
3	-EL	Negative Limit Switch
4	ORG	Origin Switch
5	DIR	DIR output
6	Reserved	
7	PCS	PCS signal input
8	ERC	ERC pin output
9	EZ	Index signal
10	Reserved	
11	Latch	Latch signal input
12	SD	Slow Down signal input
13	INP	In-Position signal input
14	SVON	Servo-ON output status

Remarks

This property is read only.
In VC++,C++ Builder
Variant is a VARIANT of VT_I2.

Data Type

VARIANT.

Axis#.ManualInputMode Property

Returns/sets pulser input mode and operation mode for one axis. The '#' can be '0' to '3'. It represents the axis identification.

Syntax

object.Axis#.ManualInputMode [=number]

Settings

Value	Description
0	1X AB PHASE NON REVERS Mode
1	2X AB PHASE NON REVERS Mode
2	4X AB PHASE NON REVERS Mode
3	CW/CCW NON REVERS Mode
4	1X AB PHASE REVERSE Mode
5	2X AB PHASE REVERSE Mode
6	4X AB PHASE REVERSE Mode
7	CW/CCW REVERSE Mode

Data Type

Integer.

Axis#.OrgMode Property

Returns/sets Configure the logic of origin switch for one axis. The '#' can be '0' to '3'. It represents the axis identification.

Syntax

object.Axis#.OrgMode [=number]

Settings

Value	Description

0	active low
1	active high

Data Type

Integer.

Axis#.Outputmode Property

Returns/sets Set the mode for pulse command output for one axis. The '#' can be '0' to '3'. It represents the axis identification.

Syntax

object. Axis#.Outputmode[=Integer]

Settings

Value	Description
0	OUT_FALLINF_DIR_HIGH
2	OUT_RISING_DIR_HIGH
3	OUT_FALLINF_DIR_LOW
4	OUT_RISING_DIR_LOW
5	CW_CCW_FALLING
6	CW_CCW_RISING

Data Type

Integer.

Axis#.ScalingFactor Property

Returns/sets resolution ratio for one axis. The '#' can be '0' to '3'. It represents the axis identification.

Syntax

object.Axis#.ScalingFactor [=value]

Settings

resolution ratio for the axis.

Data Type

Double.

Axis#.SdMode Property

Returns/sets active logic and latch control of SD signal input from mechanical system for one axis. The '#' can be '0' to '3'. It represents the axis identification.

Syntax

object.Axis#.SdMode [=number]

Settings

Value	Description
0	disable
1	enable active low not latch
2	enable active high not latch
3	enable active low latch
4	enable active high latch
5	enable active low not latch
6	enable active high not latch
7	enable active low latch
8	enable active high latch

Data Type

Integer.

Axis#.Svon Property

Returns/sets the High/Low state of general purpose output pin SVON for one axis. The '#' can be '0' to '3'. It represents the axis identification.

Syntax

object.Axis#.Svon [=number]

Settings

Value	Description
0	SVON LOW
1	SVON HIGH

Data Type

Integer.

DeviceIndex Property

Returns/sets a value that determines the card index of PCIM 1002E .

Syntax

object. DeviceIndex [= number]

Data Type

Integer.

IntControl Property

Returns/sets the status of an interrupt.

Syntax

object. IntControl [= Boolean]

Settings

Value	Description
True	enable int flag.
False	disable int flag.

Data Type

Boolean.

IntStatus Property

Returns/sets the status of an interrupt.

Syntax

object. IntStatus [= Integer]

Settings

Value	Description
1	enable interrupt.
0	disable interrupt.

Data Type

Integer.

LoadConfig Property

Returns/sets a value that determines the mode of opening the device .

Syntax

object. LoadConfig [= Boolean]

Settings

Value	Description
True	Load the same configuration as “Motion Creator”.
False	Do not load the configuration from “Motion Creator”.

Data Type

Boolean.

OpenMode Property

Returns/sets a value that determines the mode of opening the device .

Syntax

object.**OpenMode** [= number]

Settings

Value	Description
0	Automatically open device when the control was created
1	Do not open device when the control was created. Must call the

Open method to open device. (Manual)

Data Type

Integer.

Methods

BuildCompareFunction Method

Syntax

Function *object*.BuildCompareFunction (axis As Integer, start As Double, end As Double, interval As Double, device As Integer) As Integer

Arguments

axis As Integer : axis number designated to move or stop.

Start, end, interval As Double : start point, end point, interval of compare function.

Device As Integer : reload device for comparator data.

0	RAM & interrupt
1	FIFO

Return Value

0 if the function is successful.

Remarks

Build compare data via compare constant interval.

BuildCompareTable Method

Syntax

Function *object*.BuildCompareTable (axis As Integer, Table as Variant, size as Long, Device as Integer) As Integer

Arguments

axis As Integer : axis number designated to move or stop.

Table as variant : compare table.

In VC++, C++ Builder

Table is a variant of VT_ARRAY|VT_I4.

size As Long : the elements number you want to build in the compare table.

Device As Integer : the reload device for comparator data.

0 : RAM & interrupt

1 : FIFO

Return Value

0 if the function is successful.

Remarks

Build compare table.

EMGStop Method

Syntax

Function *object*. EMGStop (axis As Integer) As Integer

Arguments

axis As Integer : axis number designated to move or stop.

Return Value

0 if the function is successful.

Remarks

This method will make axis immediately stop.

FixSpeedRange Method

Syntax

Function *object*.FixSpeedRange (axis As Integer, maxvel As Double) As Integer

Arguments

axis As Integer : axis number designated to move or stop.

Maxvel As Double : maximum velocity in units of pulse per second

Return Value

0 if the function is successful.

Remarks

This method will define the speed range.

GetCurrentSpeed Method

Syntax

Function *object*.GetCurrentSpeed (axis As Integer, speed As Double) As Integer

Arguments

axis As Integer : axis number designated to move or stop.

speed As Double : current pulse output rate.

Return Value

0 if the function is successful.

Remarks

get the current pulse output rate of specified axis
In VC++,C++ Builder
speed is the address to save the data.

GetDIO Method

Syntax

Function *object*.GetDIO (DiData As Variant) As Integer

Arguments

DiData As Variant : a 16-bit input value

In VC++, C++ Builder

DiData is a variant(VT_I2) address to save data.

Return Value

0 if the function is successful.

Remarks

The bit0 ~ bit5 for channel 0~5 respectively.

GetPosition Method

Syntax

Function *object*. GetPosition (axis As Integer, pos As Double) As Integer

Arguments

axis As Integer : axis number designated to move or stop.

pos As Double : actual position.

Return Value

0 if the function is successful.

Remarks

get the actual position
In VC++,C++ Builder
Pos is the address to save the position.

HomeMove Method

Syntax

Function *object*.HomeMove (axis As Integer, svel As Double, mvel As Double, accel As Double) As Integer

Arguments

axis As Integer : axis number designated to move or stop.

0	axis 0
1	axis 1
2	axis 2
3	axis3

svel As Double : starting velocity in units of pulse per second.

mvel As Double : maximum velocity in units of pulse per second.

accel As Double : specified acceleration time in units of seconds.

Return Value

0 if the function is successful.

Remarks

Begin a home return move.

Open Method

Syntax

Function *object*.**Open (***[ErrMsgBox As Variant]***)** As Boolean

Arguments

[ErrMsgBox As Variant]

It is optional and boolean type. Default value is False

True, It will popup an error message dialog box when the opening device is failed.

False, It will give a DAQError event instead of popping up dialog when the opening device is failed.

Return Value

True if the function is successful; otherwise False.

Remarks

This method will be use when the OpenMode property is Manual.

Note

In VC++, *ErrMsgBox* is a VARIANT of VT_I2.

PulserHomeMove Method

Syntax

Function *object*.PulserHomeMove (axis As Integer, HomeType As Integer, SpeedLimit As Double) As Integer

Arguments

axis As Integer : axis number designated to move or stop.

HomeType As Integer : home move type

0	Command Origin
1	ORG pin

SpeedLimit As Double : the maximum speed in pulser move.

Return Value

0 if the function is successful.

Remarks

Manual pulser home move.

PulserPMove Method

Syntax

Function *object*.PulserPMove (axis As Integer, dis As Double, SpeedLimit As Double) As Integer

Arguments

axis As Integer : axis number designated to move or stop.

Dis As Double : relative distance to move.

SpeedLimit As Double : the maximum speed in pulser move.

Return Value

0 if the function is successful.

Remarks

Manual pulser P-move.

PulserVMove Method

Syntax

Function *object*.PulserVMove (axis As Integer, SpeedLimit As Double) As Integer

Arguments

axis As Integer : axis number designated to move or stop.

SpeedLimit As Double : the maximum speed in pulser move.

Return Value

0 if the function is successful.

Remarks

Manual pulser V-move.

SDStop Method

Syntax

Function *object*.SDStop(axis As Integer, tdec As Double) As Integer

Arguments

axis As Integer : axis number designated to move or stop.

tdec As Double : specified acceleration time in units of seconds.

Return Value

0 if the function is successful.

Remarks

This method will make axis decelerate then stop.

SetAutoCompare Method

Syntax

Function *object*.SetAutoCompare (axis As Integer, SelectSrc As Integer) As Integer

Arguments

axis As Integer : axis number designated to move or stop.

SelectSrc As Integer : comparing source counter

0	disable auto compare
1	use FIFO
2	use compare function
3	use compare table

Return Value

0 if the function is successful.

Remarks

Set comparing data source for auto loading.

SetConfigFlie Method

Syntax

Function *object*.SetConfigFlie (filename As String) As Integer

Arguments

filename As String : the configuration file name.

Return Value

0 if the function is successful.

Remarks

Save the configuration.

SetDO Method

Syntax

Function *object*.SetDO (Ch_No As Integer, DoData As Integer) As Integer

Arguments

Ch_No As Integer : channel number 0~5

DoData As Long : On-Off value for output

Return Value

0 if the function is successful.

Remarks

Set a value to PCIM 2004E's digital output channel.

SetGeneralComparator Method

Syntax

Function *object*.SetGeneralComparator (axis As Integer, CmpSrc As Integer, CmpMethod As Integer, CmpAction As Integer, Data As Double) As Integer

Arguments

axis As Integer : axis number designated to move or stop.

CmpSrc As Integer : comparing source counter

0	Command Counter
1	Feedback Counter
2	Error Counter
3	General Counter

CmpMethod As Integer : comparing method

0	No compare
1	CmpValue=Counter
2	CmpValue=Counter(+Dir)
3	CmpValue=Counter(-Dir)
4	CmpValue>Counter
5	CmpValue<Counter

CmpAction As Integer : the reaction mode when comparison comes into exist

0	INT only
1	Immediately stop
2	Slow down than stop
3	Speed change

Data As Double :Comparing value.

Return Value

0 if the function is successful.

Remarks

Set the general-purposed comparator.

SetGeneralCounter Method

Syntax

Function *object*.SetGeneralCounter (axis As Integer, cntsrc As Integer, cntvalue As Double) As Integer

Arguments

axis As Integer : axis number designated to move or stop.

cntvalue As Double : general counter value.

cntsrc As Integer : Source of general counter

0	command
1	EA/EB
2	PA/PB
3	CLK/2

Return Value

0 if the function is successful.

Remarks

Set the general counter.

SetPosition Method

Syntax

Function *object*. SetPosition (axis As Integer, pos As Double) As Integer

Arguments

axis As Integer : axis number designated to move or stop.

pos As Double : actual position.

Return Value

0 if the function is successful.

Remarks

Set the actual position.

SetPulserIptMode Method

Syntax

Function *object*.SetPulserIptMode (axis As Integer, inputmode As Integer, inverse As Integer) As Integer

Arguments

axis As Integer : axis number designated to move or stop.

inputmode As Integer : setting of manual pulser input mode from PA and PB pins

0	1X AB PHASE
1	2X AB PHASE
2	4X AB PHASE
3	CW/CCW

inverse As Integer : Reverse the moving direction from pulse direction

0	no inverse
1	inverse

Return Value

0 if the function is successful.

Remarks

Set the input signal modes of pulser.

SetTriggerComparator Method

Syntax

Function *object*.SetTriggerComparator (axis As Integer, CmpSrc As Integer, CmpMethod As Integer, Data As Double) As Integer

Arguments

axis As Integer : axis number designated to move or stop.

CmpSrc As Integer : comparing source counter

0	Command Counter
1	Feedback Counter
2	Error Counter
3	General Counter

CmpMethod As Integer : comparing method

0	No compare
1	CmpValue=Counter
2	CmpValue=Counter(+Dir)
3	CmpValue=Counter(-Dir)
4	CmpValue>Counter
5	CmpValue<Counter

Data As Double :Comparing value

Return Value

0 if the function is successful.

Remarks

Set the trigger comparator.

SetTriggerType Method

Syntax

Function *object*.SetTriggerType (axis As Integer, TriggerType As Integer) As Integer

Arguments

axis As Integer : axis number designated to move or stop.

TriggerType As Integer : comparing source counter

0	One shoot
1	high

Return Value

0 if the function is successful.

Remarks

Set the trigger output type.

StartAArc2 Method

Syntax

Function *object*.StartRArc2 (axisarray As Variant, Cx As Double, Cy As Double, Ex As Double, Ey As Double, DIR As Integer, maxvel As Double)As Integer

Arguments

Cx, Cy As Double : absolute position of center.

Ex, Ey As Double : absolute position of axis to move.

DIR As Integer : the direction of arc, 0 : CW, 1 : CCW

maxvel As Double : maximum velocity in units of pulse per second.

axisarray As Variant : axis number designated to move or stop.

In VC++, C++ Builder

axisarray is a Variant of VT_ARRAY|VT_I2

Return Value

0 if the function is successful.

Remarks

Begin an absolute circular interpolation for any two axes.

StartRArc2 Method

Syntax

Function *object*.StartRArc2 (axisarray As Variant, OffsetCx As Double, OffsetCy As Double, OffsetEx As Double, OffsetEy As Double, DIR As Integer, maxvel As Double)As Integer

Arguments

OffsetCx, OffsetCy As Double : offset to center.

OffsetEx, OffsetEy As Double : offset to end of arc.

DIR As Integer : the direction of arc, 0 : CW, 1 : CCW

maxvel As Double : maximum velocity in units of pulse per second.

axisarray As Variant : axis number designated to move or stop.

In VC++, C++ Builder

axisarray is a Variant of VT_ARRAY|VT_I2

Return Value

0 if the function is successful.

Remarks

Begin a relative circular interpolation for any two axes.

StartSALine2 Method

Syntax

Function *object*.StartSALine2 (axisarray As Variant, posX As Double, posY As Double, strvel As Double, maxvel As Double, tacc As Double, tdec As Double, svacc As Double, svdec As Double)As Integer

Arguments

PosX, posY As Double : the absolute position to move.

strvel As Double : starting velocity in units of pulse per second.

maxvel As Double : maximum velocity in units of pulse per second.

tacc, tdec, svacc, svdec As Double : specified acceleration time in units of seconds.

axisarray As Variant : axis number designated to move or stop.

In VC++, C++ Builder

axisarray is a Variant of VT_ARRAY|VT_I2

Return Value

0 if the function is successful.

Remarks

Begin an absolute 2-axis linear interpolation for any two axes with S-curve Profile.

StartSALine3 Method

Syntax

Function *object*.StartSALine3 (axisarray As Variant, posX As Double, posY As Double, posZ As Double, strvel As Double, maxvel As Double, tacc As Double, tdec As Double, svacc As Double, svdec As Double)As Integer

Arguments

PosX, posY, posZ As Double : the absolute position to move.

strvel As Double : starting velocity in units of pulse per second.

maxvel As Double : maximum velocity in units of pulse per second.

tacc, tdec, svacc, svdec As Double : specified acceleration time in units of seconds.

axisarray As Variant : axis number designated to move or stop.

In VC++, C++ Builder

axisarray is a Variant of VT_ARRAY|VT_I2

Return Value

0 if the function is successful.

Remarks

Begin an absolute 3-axis linear interpolation for any three axes with S-curve Profile.

StartSALine4 Method

Syntax

Function *object*.StartSALine4 (posX As Double, posY As Double, posZ As Double, posU As Double, strvel As Double, maxvel As Double, tacc As Double, tdec As Double, svacc As Double, svdec As Double)As Integer

Arguments

PosX, posY, posZ, posU As Double : the absolute position to move.

strvel As Double : starting velocity in units of pulse per second.

maxvel As Double : maximum velocity in units of pulse per second.

tacc, tdec, svacc, svdec As Double : specified acceleration time in units of seconds.

Return Value

0 if the function is successful.

Remarks

Begin an absolute 4-axis linear interpolation for four axes with S-curve Profile.

StartSAMove Method

Syntax

Function *object*.StartSAMove (axis As Integer, pos As Double, strvel As Double, maxvel As Double, tacc As Double, tdec As Double, svacc As Double, svdec As Double) As Integer

Arguments

axis As Integer : axis number designated to move or stop.

pos As Double : the absolute position to move.

strvel As Double : starting velocity in units of pulse per second.

maxvel As Double : maximum velocity in units of pulse per second.

tacc, tdec, svacc, svdec As Double : specified acceleration time in units of seconds.

Return Value

0 if the function is successful.

Remarks

Begin an absolute S-Curve profile move.

StartSRLine2 Method

Syntax

Function *object*.StartSALine2 (axisarray As Variant, disX As Double, disY As Double, strvel As Double, maxvel As Double, tacc As Double, tdec As Double, svacc As Double, svdec As Double)As Integer

Arguments

disX, disY As Double : the relative distance to move.

strvel As Double : starting velocity in units of pulse per second.

maxvel As Double : maximum velocity in units of pulse per second.

tacc, tdec, svacc, svdec As Double : specified acceleration time in units of seconds.

axisarray As Variant : axis number designated to move or stop.

In VC++, C++ Builder

axisarray is a Variant of VT_ARRAY|VT_I2

Return Value

0 if the function is successful.

Remarks

Begin a relative 2-axis linear interpolation for any two axes with S-curve Profile.

StartSRLine3 Method

Syntax

Function *object*.StartSALine3 (axisarray As Variant, disX As Double, disY As Double, disZ As Double, strvel As Double, maxvel As Double, tacc As Double, tdec As Double, svacc As Double, svdec As Double)As Integer

Arguments

disX, disY, disZ As Double : the relative distance to move.

strvel As Double : starting velocity in units of pulse per second.

maxvel As Double : maximum velocity in units of pulse per second.

tacc, tdec, svacc, svdec As Double : specified acceleration time in units of seconds.

axisarray As Variant : axis number designated to move or stop.

In VC++, C++ Builder

axisarray is a Variant of VT_ARRAY|VT_I2

Return Value

0 if the function is successful.

Remarks

Begin a relative 3-axis linear interpolation for any three axes with S-curve Profile.

StartSRLine4 Method

Syntax

Function *object*.StartSALine4 (disX As Double, disY As Double, disZ As Double, disU As Double, strvel As Double, maxvel As Double, tacc As Double, tdec As Double, svacc As Double, svdec As Double)As Integer

Arguments

disX, disY, disZ, disU As Double : the relative distance to move.

strvel As Double : starting velocity in units of pulse per second.

maxvel As Double : maximum velocity in units of pulse per second.

tacc, tdec, svacc, svdec As Double : specified acceleration time in units of seconds.

Return Value

0 if the function is successful.

Remarks

Begin a relative 4-axis linear interpolation for four axes with S-curve Profile.

StartSRMove Method

Syntax

Function *object*.StartSRMove (axis As Integer, dis As Double, strvel As Double, maxvel As Double, tacc As Double, tdec As Double, svacc As Double, svdec As Double) As Integer

Arguments

axis As Integer : axis number designated to move or stop.

dis As Double : the relative distance to move.

strvel As Double : starting velocity in units of pulse per second.

maxvel As Double : maximum velocity in units of pulse per second.

tacc, tdec, svacc, svdec As Double : specified acceleration time in units of seconds.

Return Value

0 if the function is successful.

Remarks

Begin a relative S-Curve profile move.

StartTALine2 Method

Syntax

Function *object*.StartTALine2 (axisarray As Variant, posX As Double, posY As Double, strvel As Double, maxvel As Double, tacc As Double, tdec As Double)As Integer

Arguments

PosX, posY As Double : the absolute position to move.

strvel As Double : starting velocity in units of pulse per second.

maxvel As Double : maximum velocity in units of pulse per second.

tacc, tdec As Double : specified acceleration time in units of seconds.

axisarray As Variant : axis number designated to move or stop.

In VC++, C++ Builder

axisarray is a Variant of VT_ARRAY|VT_I2

Return Value

0 if the function is successful.

Remarks

Begin an absolute 2-axis linear interpolation for any two axes.

StartTALine3 Method

Syntax

Function *object*.StartTALine3 (axisarray As Variant, posX As Double, posY As Double, posZ As Double, strvel As Double, maxvel As Double, tacc As Double, tdec As Double)As Integer

Arguments

PosX, posY, posZ As Double : the absolute position to move.

strvel As Double : starting velocity in units of pulse per second.

maxvel As Double : maximum velocity in units of pulse per second.

tacc, tdec As Double : specified acceleration time in units of seconds.

axisarray As Variant : axis number designated to move or stop.

In VC++, C++ Builder

axisarray is a Variant of VT_ARRAY|VT_I2

Return Value

0 if the function is successful.

Remarks

Begin an absolute 3-axis linear interpolation for any three axes.

StartTALine4 Method

Syntax

Function *object*.StartTALine4 (posX As Double, posY As Double, posZ As Double, posU As Double, strvel As Double, maxvel As Double, tacc As Double, tdec As Double)As Integer

Arguments

PosX, posY, posZ, posU As Double : the absolute position to move.

strvel As Double : starting velocity in units of pulse per second.

maxvel As Double : maximum velocity in units of pulse per second.

tacc, tdec As Double : specified acceleration time in units of seconds.

Return Value

0 if the function is successful.

Remarks

Begin an absolute 4-axis linear interpolation for four axes.

StartTAMove Method

Syntax

Function *object*.StartTAMove (axis As Integer, pos As Double, strvel As Double, maxvel As Double, tacc As Double, tdec As Double) As Integer

Arguments

axis As Integer : axis number designated to move or stop.

pos As Double : the absolute position to move.

strvel As Double : starting velocity in units of pulse per second.

maxvel As Double : maximum velocity in units of pulse per second.

tacc, tdec As Double : specified acceleration time in units of seconds.

Return Value

0 if the function is successful.

Remarks

Begin an absolute trapezoidal profile move.

StartTRLine2 Method

Syntax

Function *object*.StartTRLine2 (axisarray As Variant, disX As Double, disY As Double, strvel As Double, maxvel As Double, tacc As Double, tdec As Double)As Integer

Arguments

disX, disY As Double : the relative distance to move.

strvel As Double : starting velocity in units of pulse per second.

maxvel As Double : maximum velocity in units of pulse per second.

tacc, tdec As Double : specified acceleration time in units of seconds.

axisarray As Variant : axis number designated to move or stop.

In VC++, C++ Builder

axisarray is a Variant of VT_ARRAY|VT_I2

Return Value

0 if the function is successful.

Remarks

Begin a relative 2-axis linear interpolation for any two axes.

StartTRLine3 Method

Syntax

Function *object*.StartTRLine3 (axisarray As Variant, disX As Double, disY As Double, disZ As Double, strvel As Double, maxvel As Double, tacc As Double, tdec As Double)As Integer

Arguments

disX, disY, disZ As Double : the relative distance to move.

strvel As Double : starting velocity in units of pulse per second.

maxvel As Double : maximum velocity in units of pulse per second.

tacc, tdec As Double : specified acceleration time in units of seconds.

axisarray As Variant : axis number designated to move or stop.

In VC++, C++ Builder

axisarray is a Variant of VT_ARRAY|VT_I2

Return Value

0 if the function is successful.

Remarks

Begin a relative 3-axis linear interpolation for any three axes.

StartTRLine4 Method

Syntax

Function *object*.StartTRLine4 (disX As Double, disY As Double, disZ As Double, disU As Double, strvel As Double, maxvel As Double, tacc As Double, tdec As Double)As Integer

Arguments

disX, disY, disZ, disU As Double : the relative distance to move.

strvel As Double : starting velocity in units of pulse per second.

maxvel As Double : maximum velocity in units of pulse per second.

tacc, tdec As Double : specified acceleration time in units of seconds.

Return Value

0 if the function is successful.

Remarks

Begin a relative 4-axis linear interpolation for four axes.

StartTRMove Method

Syntax

Function *object*.StartTRMove (axis As Integer, dis As Double, strvel As Double, maxvel As Double, tacc As Double, tdec As Double) As Integer

Arguments

axis As Integer : axis number designated to move or stop.

dis As Double : the relative distance to move.

strvel As Double : starting velocity in units of pulse per second.

maxvel As Double : maximum velocity in units of pulse per second.

tacc, tdec As Double : specified acceleration time in units of seconds.

Return Value

0 if the function is successful.

Remarks

Begin a relative trapezoidal profile move.

SVMove Method

Syntax

Function *object*.SVMove(axis As Integer, strvel As Double, maxvel As Double, tacc As Double, svacc As Double) As Integer

Arguments

axis As Integer : axis number designated to move or stop.

strvel As Double : starting velocity in units of pulse per second.

maxvel As Double : maximum velocity in units of pulse per second.

tacc, svacc As Double : specified acceleration time in units of seconds.

Return Value

0 if the function is successful.

Remarks

This method will accelerate an axis to a constant velocity with S-curve profile.

TVMove Method

Syntax

Function *object*.TVMove(axis As Integer, strvel As Double, maxvel As Double, Tacc As Double) As short

Arguments

axis As Integer : axis number designated to move or stop.

strvel As Double : starting velocity in units of pulse per second.

maxvel As Double : maximum velocity in units of pulse per second.

Tacc As Double : specified acceleration time in units of seconds.

Return Value

0 if the function is successful.

Remarks

This method will accelerate an axis to a constant velocity with trapezoidal profile.

UnFixSpeedRange Method

Syntax

Function *object***.**UnFixSpeedRange (axis As Integer) As Integer

Arguments

axis As Integer : axis number designated to move or stop.

Return Value

0 if the function is successful.

Remarks

This method will release the speed range constraint.

VChange Method

Syntax

Function *object***.**VChange(axis As Integer, newvel As Double, tacc As Double) As Integer

Arguments

axis As Integer : axis number designated to move or stop.

newvel As Double : maximum velocity in units of pulse per second.

tacc As Double : specified acceleration time in units of seconds.

Return Value

0 if the function is successful.

Remarks

This method will Change speed instantly.

Events

OnIntEvent Event

Syntax

Sub *ControlName_* OnIntEvent**(** *axis As Integer,* int_status *As Long,* error_status As Long **)**

Arguments

axis As Intege : the axis that generates interrupts.
int_status *As Long :* the interrupt status.
error_status As Long : the error status.

Remarks

This event will occur when an interrupt or error is generated.

Note

User must enable the interrupt and set the interrupt factor.
The interrupt will be disabled when you use auto compare.